REMARKS

Applicants respectfully request reconsideration and allowance of the subject application. Claims 1-40 are pending in this application.

REJECTIONS UNDER 35 USC §101

Claims 8-13, 14-18, 23-30, and 38-40 stand rejected under 35 USC §101 as being directed to non-statutory subject matter, namely, abstract ideas. Applicants have amended claims 8, 14, 23, and 38, (and claims 9-13, 15-18, 24-30, and 39-40 depending therefrom) and respectfully request reconsideration and withdrawal of the rejection.

Applicants submit that claims 8-13, 14-18, 23-30, and 38-40 properly recite statutorily allowable subject matter. As amended, claims 8-13, 14-18, 23-30, and 38-40 are proper statutory claims under 35 USC 101 because these claims are properly "limited by the language in the claims to a practical application within the technological arts." § MPEP 2106.IV.B.2.(b); *citing Diamond v. Diehr*, 450 U.S. 175, 183-4, 209 USPQ 1, 6 (1981). "Only when the claim is devoid of any limitation to a practical application in the technological arts should it be rejected under 35 USC 101." § MPEP 2106.IV.(e).

More specifically, amended claim 8 recites, in relevant part, "A hashing *architecture for determining whether an input value matches any of a plurality of target values*, comprising: ... a comparator coupled to receive the hash result *and to determine which of the plurality of target values to compare to the input value*." (emphasis added). Since claim 8 properly recites a practical application, it should not be rejected under 35 USC §101.

Amended claim 14 recites, in relevant part, "A *method of determining whether an input value matches any of a plurality of target values*, comprising ... *determining based on the hash result which of the plurality of target values to compare to the input value*." (emphasis added). Because claim 14 properly recites a practical application, it should not be rejected under 35 USC §101.

Claim 23 recites, in relevant part, "*A method of determining whether an input security identifier matches one or more of a plurality of target security identifiers*, the method comprising ... comparing the input security identifier to at least one of the plurality of target security identifiers that corresponds to a portion of the result hash value having a particular value *to determine whether a match exists*." Again, since claim 23 properly recites a practical application (i.e. *determining whether a match exists*), it should not be rejected under 35 USC §101.

Claim 38 recites, in relevant part, "A method *for determining whether an input value matches a target value,* comprising: for each sub-hash in a plurality of sub-hashes that can be used together to generate a hash result *for determining whether the input value matches the target value* [.]" Because claim 38 properly recites a practical application it should not be rejected under 35 USC §101.

For the foregoing reasons, Applicants respectfully request reconsideration and withdrawal of the rejections of claims 8-13, 14-18, 23-30, and 38-40 under 35 USC §101 as being non-statutory.

REJECTIONS UNDER 35 U.S.C. § 103

In the Office Action mailed December 5, 2005, the Office maintained the rejections under 35 U.S.C. §103(a) on the same grounds as articulated in the first Office Action mailed on April 22, 2004. Accordingly, in the following analysis, specific references are made to the relevant portions of the first Office Action mailed on April 22, 2004.

Claims 1-7 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,505,191 to Baclawski (hereinafter "Baclawski") in view of U.S. Patent No. 5,852,821 to Chen et al. (hereinafter "Chen") and further in view of Applicant's Admitted Prior Art (hereinafter "AAPA"). Applicants respectfully submit that claims 1-7 are not obvious over Baclawski in view of Chen and AAPA.

In general, Applicants teach methods and systems for determining whether an input value matches one or more of a plurality of target values. In one embodiment, a method includes generating a hash key based on the input value, and separating the hash key into a plurality of portions. (Specification, p. 8, lines 7-8). The plurality of portions are used to index into a plurality of sub-hashes.

More specifically, as best shown in Applicants' Figure 3, for a system having two sub-hashes 160, 162, the input value SID 132 is hashed and then separated into two portions (*e.g.* a high portion *h* and a low portion *l*).

(Specification, p. 8, lines 7-19). The value of each of the high and low portions *h*, *l* is then determined, and used as an index into a corresponding one of the sub-hashes 160, 162. (Specification, p. 8, lines 15-19).

Each location in the sub-hashes 160, 162 contains a multiple-bit value, each bit corresponding to one of the plurality of target values. (Specification, p. 8, lines 20-21). A plurality of values are identified from the plurality of sub-hashes based on the indexing. The method further includes combining the plurality of values to generate a hash result, wherein each bit in the hash result corresponds to one of the plurality of target values. In one embodiment, the plurality of values are combined using combinatorial logic to generate the hash result. (Specification, p. 9, lines 5-8). For each bit in the hash result that is set, the input value is compared to the corresponding target value to determine whether the values match.

Baclawski (U.S. 6,505,191)

Baclawski describes computerized information retrieval systems and methods employing hypertext linkage analysis (3:41-58). According to Baclawski, a query from a user is transmitted to a front end computer, which forwards the query to a home node of a search engine. The home node parses the query into one or more elementary queries and schedules the elementary queries for processing. Each elementary query can be one of a number of types, including an index query, a link query, or an object query. To process an index query or link query, the home node extracts features from the index query or link query, fragments the extracted features into feature fragments, and hashes these features. (4:45:54). The feature fragments are used to find matching fragments in the database, so they are also called probes. (8:7-9). The feature fragments are hashed, and each hashed feature fragment is transmitted to one index node on the network. (4:54-56). Each index node on the network that receives a hashed feature fragment uses the hashed feature fragment of the index query or link query to perform a search on its respective partition of the database. (4:56-59). The home node then receives the search results and processes the results for each elementary query according to the specifications in the query. (4:59-5:2).

Baclawski does not disclose, teach, or fairly suggest the methods taught by Applicants. For example, Baclawski does not teach or fairly suggest a method that includes separating a hash key into multiple portions, and then using the value of each of the portions as an index into a corresponding sub-hash of a plurality of sub-hashes, wherein each location in the sub-hashes 160, 162 contains a multiple-bit value, each bit corresponding to one of the plurality of target values. (Specification, p. 8, lines 15-21).

Baclawski is cited by the Examiner as teaching "indexing into each of a plurality of sub-hashes (index nodes) using one of the plurality of portions (see for example; col 9 ln 26-43)." (Office Action mailed April 22, 2004, p. 3, lines 4-5). Applicants respectfully submit that the Baclawski does not teach or fairly suggest the indexing taught by Applicants. According to Baclawski, a home node 105 hashes each fragment of the query and transmits the hashed fragment to an index node 112, which maintains an index to the data in the local database. (8:61-66). At the index node 112, a comparison is made between the hashed fragment and the terms in the index, and "all matches between the hashed probes and the local hash table of index terms are returned or gathered to the home node 107." (9:30-33).

There is no teaching or suggestion in Baclawski of separating a hash key into multiple portions, and then using the value of each of the portions as an index into a corresponding sub-hash of a plurality of sub-hashes, wherein each location in the sub-hashes 160, 162 contains a multiple-bit value, each bit corresponding to one of the plurality of target values. (Specification, p. 8, lines 15-21). According to Baclawski, the comparisons are made between the hashed fragments and the terms in the index. Thus, Applicants respectfully submit that Baclawski is another example of a prior art comparison technique that does not that does not afford the advantages of reduced operations and reduced memory requirements realizeable using embodiments of Applicants' invention. (Specification, p. 1, line 13, to p. 2, line 17).

Furthermore, as noted by the Examiner, Baclawski does not teach "generating a hash key based on the input value and separating the hash key into a plurality of portions." (Office Action, p. 3, lines 6-18). In fact, Baclawski teaches an opposite approach: separating the input value into fragments, and then hashing each of the fragments. (4:51-54;8:61-66). The Examiner dismisses this distinction on grounds that Applicants "did not explicitly state any reason or purpose for such means other than for generating an index to lookup values and the means disclosed by Baclawski is just as efficient." (Office Action, p. 3, lines 15-17). Applicants respectfully disagree. Baclawski's method is inherently less efficient because at least twice as many hashing operations must be performed than are required using Applicants' method. In other words, because the hashing of Baclawski is performed after the input value is separated into a plurality of fragments, more hashing operations are required rather by than simply hashing the single input value itself prior to separation, as taught by Applicants. Thus, Baclawski teaches an opposite approach that is less efficient and therefore less desirable, and is not a mere obvious "alternate approach."

The Examiner also notes that Baclawski fails to teach or fairly suggest two other aspects of Applicants' invention, namely, "combining a plurality of values to generate a hash result and comparing the input value to corresponding target value to determine whether the values match." (Office Action, p. 3, lines 18-20). As described below, Applicants respectfully submit that these deficiencies are not remedied by the teachings of the other cited references.

Chen (U.S. 5,852,821)

Chen describes high-speed data base query methods and apparatus. In Chen, an index of bit vectors is created by accessing one of the values stored in a database and assigning each bit of the bit pattern for that value, from the most significant bit to the least significant bit, to a unique position in successive bit vectors (see, col. 4, lines 31-39). This accessing and assigning is repeated for each remaining value to form an index of bit vectors for the values (see, col. 4, lines 39-42). In order to search the index thus created for retrieving and/or reconstructing those data values greater than a search value, these bit vectors are used, in conjunction with the search value, to generate multiple answer vectors (see, col. 5, line 31- col. 6, line 13).

Applicants respectfully submit that Chen fails to remedy the above-noted deficiencies of Baclawski. Specifically, Chen fails to teach or fairly suggest: (1)

separating a hash key into multiple portions, and then using the value of each of the portions as an index into a corresponding sub-hash of a plurality of sub-hashes, wherein each location in the sub-hashes contains a multiple-bit value, each bit corresponding to one of the plurality of target values; (2) combining a plurality of values to generate a hash result; (3) generating the hash result by combining the plurality of values using combinatorial logic; and (4) comparing the input value to corresponding target value to determine whether the values match.

With respect to combining a plurality of values to generate a hash result, the Examiner asserts that "Chen discloses a second means of matching input value with a target value wherein a plurality of values are combined to generate a result (see for example; col. 5 lines 21-30)." Applicants respectfully disagree with this characterization of Chen.

Chen does not discuss hashing or hash results, much less combining a plurality of values to generate a hash result. The word "hash" is not even present in Chen. Absent any discussion or even mention of hashing or a hash result, Applicants respectfully submit that Chen cannot disclose or suggest combining the plurality of values to generate a hash result, wherein each bit in the hash result corresponds to one of the plurality of target values as recited in claim 1.

Furthermore, Applicants teach combining the plurality of values identified from the plurality of sub-hashes during the indexing. As there is no discussion of hashing in Chen, Applicants respectfully submit that Chen cannot disclose or suggest combining a plurality of values identified from the plurality of sub-hashes.

Applicants' Admitted Prior Art (AAPA)

Applicants' Admitted Prior Art (AAPA) describes access control systems that compare a list of multiple (n) security identifiers (SID) 106 with multiple access control elements (m) (ACE) that each includes an SID 104, requiring m x n operations. Alternately, the AAPA teaches that hashing can be used to reduce the number of operations, with relatively large memory storage requirements.

Applicants respectfully submit that AAPA fails to remedy the above-noted deficiencies of Baclawski and Chen. Specifically, AAPA fails to teach or fairly suggest: (1) separating a hash key into multiple portions, and then using the value of each of the portions as an index into a corresponding sub-hash of a plurality of sub-hashes, wherein each location in the sub-hashes contains a multiple-bit value, each bit corresponding to one of the plurality of target values; (2) combining a plurality of values to generate a hash result; and (3) generating the hash result by combining the plurality of values using combinatorial logic.

Claims 1-7

Turning now to the specific language of the claims, claim 1 recites:

> 1. (Currently Amended) One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to determine whether an input value matches any of a plurality of target values by performing acts including:
>
> generating a hash key based on the input value;
>
> separating the hash key into a plurality of portions;
>
> indexing into each of a plurality of sub-hashes using one of the plurality of portions, each location in each of the plurality of sub-hashes

containing a multiple-bit value, each bit corresponding to one of the plurality of target values;

identifying a plurality of values from the plurality of sub-hashes based on the indexing;

combining the plurality of values to generate a hash result, wherein each bit in the hash result corresponds to one of the plurality of target values; and

for each bit in the hash result that is set, comparing the input value to the corresponding target value to determine whether the values match.

As described more fully above, the cited references (Baclawski, Chen, and AAPA), either singly or in combination, do not disclose, teach, or fairly suggest the computer readable media recited in claim 1. More specifically, the cited references do not teach or suggest "generating a hash key based on the input value," and "separating the hash key into a plurality of portions." Baclawski teaches the opposite approach – separating an input value, and hashing each of the resulting portions. Chen and AAPA do not remedy this deficiency.

Similarly, the cited references fail to teach or suggest "indexing into each of a plurality of sub-hashes using one of the plurality of portions, each location in each of the plurality of sub-hashes containing a multiple-bit value, each bit corresponding to one of the plurality of target values." As noted above, Baclawski, Chen, and AAPA fail to teach these limitations of claim 1.

Baclawski, Chen, and AAPA also fail to teach or suggest "combining the plurality of values to generate a hash result, wherein each bit in the hash result corresponds to one of the plurality of target values." Again, as noted above, Baclawski, Chen, and AAPA fail to teach these limitations of claim 1.

For at least these reasons, Applicants respectfully submit that claim 1 is allowable over Baclawski in view of Chen and AAPA. Claims 2-7 depend from claim 1, and are allowable over the cited references at least due to their dependency on claim 1 and also due to additional limitations recited in those claims. Therefore, Applicants request reconsideration and withdrawal of the rejections of claims 1-7 under 35 U.S.C. §103(a).

Claims 8-13

Claims 8-13 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Chen in view of U.S. Patent No. 5,852,822 to Srinivasan et al. (hereinafter "Srinivasan"). Applicants respectfully submit that claims 8-13 are not obvious over Chen in view of Srinivasan.

Claim 8 recites in relevant part a hashing architecture for determining whether an input value matches any of a plurality of target values, comprising "a plurality of sub-hashes" and "a plurality of sub-hash indexes, each index being generated from a hash key and used to index into one of the plurality of sub-hashes, each location in each of the plurality of sub-hashes containing a multiple-bit value, each bit corresponding to one of the plurality of target values."

In the April 22 Office Action, at p. 8, Chen is recited as teaching all of the elements of claim 8 except for the "index begin generated from a hash key". As described more fully above, Applicants respectfully disagree with this characterization of Chen. Applicants respectfully submit that Chen fails to teach or fairly suggest "a plurality of sub-hashes" and "a plurality of sub-hash indexes, each index being generated from a hash key and used to index into one of the plurality of sub-hashes, each location in each of the plurality of sub-hashes containing a multiple-bit value, each bit corresponding to one of the plurality of target values."

Claim 8 further recites "a comparator coupled to receive the hash result and to determine which of the plurality of target values to compare to the input value." As described above, and as admitted by the Examiner (Office Action, p. 4, lines 10-12), there is also no teaching or suggestion in Chen of this additional limitation.

Chen discusses creating a plurality of bit vectors 44a-44d (see, col. 4, lines 45-46). The number of bit vectors created equals the length of the bit patterns for the values, so if the memory allocates 32 bits per character (or digit) for each value, then 32 bit vectors are created (see, col. 4, lines 46-50). Each value in the memory is represented by a 32-bit bit pattern, so the number 3 has a bit pattern 0...011 (see, col. 4, lines 53-56). For the first value, each bit of the bit pattern from the most significant bit to the least significant bit is assigned by the server to the first position in each of the bit vectors 44a-44d, so the most significant bit for the number 3 is assigned to the first position of the first vector 44d and the least significant bit is assigned to the first position of the last bit vector 44a (see, col. 4, lines 56-62). This is illustrated in Fig. 4 of Chen, where the bit pattern 0...011 for the value 3 is spread across the first positions in the bit vectors 44a-44d.

From the April 22 Office Action at p. 8, it appears that these bit vectors 44a-44d are being relied on as disclosing the plurality of sub-hashes of claim 8. If these bit vectors were to be the plurality of sub-hashes of claim 8 as asserted in the April 22 Office Action, then following the language of claim 8 there would need to be some plurality of sub-hash indexes used to index into one of the bit vectors disclosed in Chen. However, there is no discussion or suggestion in Chen of any such plurality of sub-hash indexes.

As stated in the cited portion of Chen, the processing includes "repeating the above-described accessing and assigning steps for each remaining value of the set to form an index of bit vectors for the values (steps 38, 40)" (see, col. 4, lines 38-41). These bit vectors form an index 46 which can be searched (see, Fig. 4 and col. 5, lines 31-34). Thus, the bit vectors themselves form the index in Chen; there is not a separate index into the bit vectors described or mentioned in Chen. As such, Applicants respectfully submit that Chen cannot disclose or suggest a plurality of sub-hash indexes used to index into one of the plurality of sub-hashes as recited in claim 8.

Furthermore, as there is no plurality of sub-hash indexes disclosed or suggested in Chen, Applicants respectfully submit that Chen cannot disclose or suggest a combiner coupled to receive values from the plurality of sub-hashes based on the plurality of sub-hash indexes as recited in claim 8.

With respect to Srinivasan, Srinivasan is not cited as curing, and does not cure, these deficiencies of Chen. Accordingly, Applicants respectfully submit that claim 8 is allowable over Chen in view of Srinivasan for at least these reasons. Claims 9-18 depend from claim 8 and are allowable due to their dependency on claim 8 and also due to additional limitations recited in those claims. Therefore, Applicants request reconsideration and withdrawal of the rejections of claims 8-13 under 35 U.S.C. §103(a).

Claims 14-18

Claims 14-18 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Chen in view of U.S. Patent No. 5,852,822 to Srinivasan et al. (hereinafter "Srinivasan"). Applicants respectfully submit that claims 14-18 are not obvious over Chen in view of Srinivasan.

Claim 14 recites in relevant part a method of determining whether an input value matches any of a plurality of target values, comprising "identifying a plurality of values from a plurality of sub-hashes by indexing into each of the plurality of sub-hashes using one of the plurality of sub-hash keys, each location in each of the plurality of sub-hashes containing a multiple-bit value, each bit corresponding to one of the plurality of target values." For the reasons set forth above in the discussion of the rejection of claim 8, neither Chen nor Srinivasan, either singly or in combination, discloses, teaches, or fairly suggests "identifying a plurality of values from a plurality of sub-hashes by indexing into each of the plurality of sub-hashes using one of the plurality of sub-hash keys, each location in each of the plurality of sub-hashes containing a multiple-bit value, each bit corresponding to one of the plurality of target values" as recited in claim 14.

Claim 14 further recites "determining based on the hash result which of the plurality of target values to compare to the input value." Again, as described more fully above, Chen nor Srinivasan, either singly or in combination, discloses, teaches, or fairly suggests "determining based on the hash result which of the plurality of target values to compare to the input value" as recited in claim 14.

Accordingly, Applicants respectfully submit that claim 14 is allowable over Chen in view of Srinivasan for at least these reasons. Claims 15-18 depend from claim 14 and are allowable due to their dependency on claim 14 and also due to additional limitations recited in those claims. Therefore, Applicants request reconsideration and withdrawal of the rejections of claims 14-18 under 35 U.S.C. §103(a).

Claims 19-22

Claims 19-37 stand rejected under 35 U.S.C. §103(a) as being unpatentable over AAPA in view of Chen and further in view of Srinivasan. Applicants respectfully submit that claims 19-37 are not obvious over AAPA in view of Chen and further in view of Srinivasan.

Claim 19 recites "generating a hash key based on the access control element security identifier" and "separating the hash key into a first portion and a second portion." As described more fully above, these limitations are not disclosed or suggested by Chen, Srinivasan, or the AAPA.

Furthermore, Applicants respectfully submit that, similar to the discussion above regarding claim 8, the combination of cited references does not disclose or suggest indexing into a first sub-hash using the first portion to identify a first sub-hash value, indexing into a second sub-hash using the second portion to identify a second sub-hash value, and combining the first sub-hash value and the second sub-hash value to generate a result value as recited in claim 19. In addition, Applicants respectfully submit that AAPA is not cited as curing, and does not cure, these deficiencies of Chen in view of Srinivasan.

For at least these reasons, Applicants respectfully submit that claim 19 is allowable over AAPA in view of Chen and further in view of Srinivasan. Claims 20-22 depend from claim 19 and are allowable over the cited references due to their dependency on claim 19 and also due to additional limitations recited in these claims. Therefore, Applicants request reconsideration and withdrawal of the rejections of claims 19-22 under 35 U.S.C. §103(a).

Claims 23-30

Claims 23-30 stand rejected under 35 U.S.C. §103(a) as being unpatentable over AAPA in view of Chen and further in view of Srinivasan. Applicants respectfully submit that claims 23-30 are not obvious over AAPA in view of Chen and further in view of Srinivasan.

Claim 23 recites a method including "generating a plurality of sub-hash indexes based on a hash key, the hash key being based on the input security identifier" and "indexing into each of a plurality of sub-hashes using a respective one of the plurality of sub-hash indexes, each location in each of the plurality of sub-hashes containing a multiple-bit value, each bit corresponding to one of the plurality of target security identifiers." As described more fully above, these limitations are not disclosed or suggested by Chen, Srinivasan, or the AAPA.

Furthermore, Applicants respectfully submit that, similar to the discussion above regarding claim 8, the combination does not disclose or suggest indexing into each of a plurality of sub-hashes using a respective one of the plurality of sub-hash indexes, and generating a result hash value by combining the plurality of values resulting from indexing into the plurality of sub-hashes as recited in claim 23. In addition, Applicants respectfully submit that AAPA is not cited as curing, and does not cure, these deficiencies of Chen in view of Srinivasan.

For at least these reasons, Applicants respectfully submit that claim 23 is allowable over AAPA in view of Chen and further in view of Srinivasan. Claims 24-30 depend from claim 23 and are allowable over the cited references due to their dependency on claim 23 and also due to additional limitations recited in these claims. Therefore, Applicants request reconsideration and withdrawal of the rejections of claims 23-30 under 35 U.S.C. §103(a).

Claims 31-37

Claims 31-37 stand rejected under 35 U.S.C. §103(a) as being unpatentable over AAPA in view of Chen and further in view of Srinivasan. Applicants respectfully submit that claims 31-37 are not obvious over AAPA in view of Chen and further in view of Srinivasan.

Claim 31 recites a system comprising "a plurality of sub-hashes, each location in each of the plurality of sub-hashes containing a multiple-bit value, each bit corresponding to one of the plurality of target security identifiers." As described more fully above, these limitations are not disclosed or suggested by Chen, Srinivasan, or the AAPA.

Furthermore, Applicants respectfully submit that, similar to the discussion above regarding claim 8, the combination of cited references does not disclose or suggest indexing into each of the plurality of sub-hashes using a respective one of the plurality of sub-hash indexes, and combining the plurality of values to generate a hash result value, wherein each bit in the hash result value corresponds to one of the plurality of security token security identifiers as recited in claim 31. In addition, Applicants respectfully submit that AAPA is not cited as curing, and does not cure, these deficiencies of Chen in view of Srinivasan.

For at least these reasons, Applicants respectfully submit that claim 31 is allowable over AAPA in view of Chen and further in view of Srinivasan. Claims 32-37 depend from claim 31 and are allowable over the cited references due to their dependency on claim 31 and also due to additional limitations recited in these claims. Therefore, Applicants request reconsideration and withdrawal of the rejections of claims 31-37 under 35 U.S.C. §103(a).

Claim 38-40

Claims 38-40 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Chen. Applicants respectfully submit that claims 38-40 are not obvious over Chen. Claim 38 recites:

> A method for determining whether an input value matches a target value, comprising:
>
> for each sub-hash in a plurality of sub-hashes that can be used together to generate a hash result for determining whether the input value matches the target value, wherein each location in each of the plurality of

sub-hashes contains a multiple-bit value, each bit corresponding to one of the plurality of target values,

(a) identifying a bit in a location of the sub-hash,

(b) identifying, in a source value, a plurality of bits corresponding to the sub-hash,

(c) comparing an identifier of the location to the plurality of bits,

(d) setting the bit if the identifier of the location matches the plurality of bits, and otherwise clearing the bit, and

(e) repeating acts (a), (b), (c), and (d) for each of a plurality of bits in the location of the sub-hash.

As described above, Chen does not disclose, teach, or fairly suggest "for each sub-hash in a plurality of sub-hashes that can be used together to generate a hash result for determining whether the input value matches the target value, wherein each location in each of the plurality of sub-hashes contains a multiple-bit value, each bit corresponding to one of the plurality of target values" as recited in claim 38.

Furthermore, in the April 22 Office Action at p. 25, Chen at col. 4 lines 30-64 is cited as disclosing the acts (a), (b), (c), (d), and (e) of claim 38. Applicants respectfully disagree with this characterization of Chen.

The cited portion of Chen discusses creating a plurality of bit vectors 44a-44d (see, col. 4, lines 45-46). The number of bit vectors created equals the length of the bit patterns for the values, so if the memory allocates 32 bits per character (or digit) for each value, then 32 bit vectors are created (see, col. 4, lines 46-50). Each value in the memory is represented by a 32-bit bit pattern, so the number 3 has a bit pattern 0...011 (see, col. 4, lines 53-56). For the first value, each bit of the bit pattern from the most significant bit to the least significant bit is assigned by the server to the first position in each of the bit vectors 44a-44d, so the most significant bit for the number 3 is assigned to the first position of the first vector

44d and the least significant bit is assigned to the first position of the last bit vector 44a (see, col. 4, lines 56-62). This is illustrated in Fig. 4 of Chen, where the bit pattern 0…011 for the value 3 is spread across the first positions in the bit vectors 44a-44d.

From the April 22 Office Action at p. 25, it appears that these bit vectors 44a-44d are being relied on as disclosing the sub-hash of claim 38. However, referring to these bit vectors of Chen, there is no comparing an identifier of the location to the plurality of bits and setting the bit if the identifier of the location matches the plurality of bits, and otherwise clearing the bit. As discussed above, the bit pattern for a particular value in Chen is spread across a particular position in the bit vectors of Chen. There is no comparing of an identifier of a location to a plurality of bits and setting a bit based on whether the location matches the plurality of bits as recited in claim 38. Rather, in Chen it is simply the bit value for the value being spread across the bit vectors. For example, as shown in Fig. 4 of Chen, the value of 3 is assigned to the bit vectors with the "11", which represents the value 3, being stored in the two vectors 44a and 44b, with all other bit vectors 44c and 44d storing "0". There is not any comparing to determine what value to store in the bit vectors of Chen – a "0" is assigned to a particular position in a particular bit vector if the corresponding position in the bit pattern is a "0", and a "1" is assigned to a particular position in a particular bit vector if the corresponding position in the bit pattern is a "1".

For at least these reasons, Applicants respectfully submit that claim 38 is allowable over Chen.

Given that claims 39-40 depend from claim 38, Applicants respectfully submit that claims 39-40 are likewise allowable over Chen for at least the reasons discussed above with respect to claim 38. Claims 39-40 depend from claim 38 and are allowable over the cited references due to their dependency on claim 38 and also due to additional limitations recited in these claims. Therefore, Applicants request reconsideration and withdrawal of the rejections of claims 38-40 under 35 U.S.C. §103(a).

Conclusion

Claims 1-40 are in condition for allowance. Applicants respectfully request reconsideration and issuance of the subject application. If there are any remaining matters that may be handled by telephone conference, the Examiner is kindly invited to telephone the undersigned.

Respectfully Submitted,

Date: March 2, 2006

By: Dale C. Barr

Dale C. Barr
Lee & Hayes, PLLC
Reg. No. 40,498
(206) 315-4001, ext 106